

Flightbooking using Micorservices

Eureka server :

<http://localhost:8761/>

localhost

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY-FLIGHTBOOKING	n/a (1)	(1)	UP (1) - 10.70.0.104/api-gateway-flightbooking:8084
BOOKING-SERVICE-FLIGHTBOOKING	n/a (1)	(1)	UP (1) - 10.70.0.104/booking-service-flightbooking:8082
FLIGHT-SERVICE-FLIGHTBOOKING	n/a (1)	(1)	UP (1) - 10.70.0.104/flight-service-flightbooking:8081

General Info

Name	Value
total-avail-memory	100mb
num-of-cpus	16
current-memory-usage	84mb (83%)
server-uptime	00:18
registered-replicas	http://localhost:8761/eureka/
unavailable-replicas	http://localhost:8761/eureka/ ,
available-replicas	

Instance Info

Name	Value
ipAddr	10.70.0.104
status	UP

Postman :

Using api gateway

Using mongoDb

Add inventory:

The screenshot shows a Postman interface with the following details:

- Request URL:** `http://localhost:8084/api/v1.0/flight/airline/inventory/add`
- Method:** POST
- Body:** JSON (selected)
- Request Body Content:**

```
1 {  
2     "airlineName": "Indigo",  
3     "fromPlace": "Chennai",  
4     "toPlace": "Bangalore",  
5     "departureTime": "2025-12-10T10:00:00",  
6     "arrivalTime": "2025-12-10T12:00:00",  
7     "tripType": "ONE-WAY",  
8     "priceOneWay": 3500,  
9     "priceRoundTrip": 0,  
10    "totalSeats": 120  
11 }
```
- Response Status:** 201 Created
- Response Body:** `6930db8025b45b0fe4b8015d`

Search flights:

The screenshot shows a Postman interface with the following details:

- Request URL:** `http://localhost:8081/api/v1.0/flight/search`
- Method:** POST
- Body:** JSON (selected)
- Request Body Content:**

```
1 {  
2     "fromPlace": "Hyderabad",  
3     "toPlace": "Bangalore",  
4     "journeyDate": "2025-12-10",  
5     "tripType": "ONE-WAY"  
6 }
```
- Response Status:** 200 OK
- Response Body:**

```
1 [  
2     {  
3         "flightId": "6930cbd877157c2d5b251bf5",  
4         "departureTime": "2025-12-10T10:00:00",  
5         "arrivalTime": "2025-12-10T12:00:00",  
6         "airlineName": "Indigo",  
7         "priceOneWay": 3500.0,  
8         "priceRoundTrip": 0.0,  
9         "tripType": "ONE-WAY",  
10        "availableSeats": 120,  
11        "fromPlace": "Hyderabad",  
12        "toPlace": "Bangalore"  
13    }  
14 ]
```

Get flight using id:

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:8084/api/v1.0/flight/6930db8025b45b0fe4b8015d`. The response status is `200 OK` with a response time of 21 ms and a body size of 380 B. The response body is a JSON object representing a flight:

```
1 {  
2   "flightId": "6930db8025b45b0fe4b8015d",  
3   "departureTime": "2025-12-10T10:00:00",  
4   "arrivalTime": "2025-12-10T12:00:00",  
5   "airlineName": "Indigo",  
6   "priceOneWay": 3500.0,  
7   "priceRoundTrip": 0.0,  
8   "tripType": "ONE-WAY",  
9   "availableSeats": 120,  
10  "fromPlace": "Chennai",  
11  "toPlace": "Bangalore"  
12 }
```

Booking using id:

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:8084/api/v1.0/booking/6930db8025b45b0fe4b8015d`. The response status is `201 Created` with a response time of 432 ms and a body size of 577 B. The response body is a JSON object representing the booking:

```
1 {  
2   "customerName": "Sravya",  
3   "email": "sravya@gmail.com",  
4   "numberofSeats": 2,  
5   "journeyDate": "2025-12-10",  
6   "seatNumbers": "12A,12B",  
7   "mealPreference": "VEG",  
8   "passengers": [  
9     { "name": "Sravya", "gender": "F", "age": 21 },  
10    { "name": "Anu", "gender": "F", "age": 22 }  
11  ]  
12 }
```

Get ticket by pnr :

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:8084/api/v1.0/booking/ticket/2FPISEYW`. The response status is `200 OK` with a response time of 51 ms and a size of 568 B. The response body is a JSON object representing a flight ticket:

```
1 {  
2   "pnr": "2FPISEYW",  
3   "customerName": "Sravya",  
4   "email": "sravya@gmail.com",  
5   "fromPlace": "Chennai",  
6   "toPlace": "Bangalore",  
7   "journeyDate": "2025-12-10",  
8   "departureTime": "2025-12-10T10:00:00",  
9   "arrivalTime": "2025-12-10T12:00:00",  
10  "airlineName": "Indigo",  
11  "numberOfSeats": 2,  
12  "seatNumbers": "12A,12B",  
13  "mealPreference": "VEG",  
14  "cancelled": false,  
15 }
```

Get history by email:

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:8084/api/v1.0/booking/history/sravya@gmail.com`. The response status is `200 OK` with a response time of 30 ms and a size of 614 B. The response body is a JSON array containing one element, which is the same flight ticket object as above:

```
1 [  
2   {  
3     "pnr": "2FPISEYW",  
4     "customerName": "Sravya",  
5     "email": "sravya@gmail.com",  
6     "fromPlace": "Chennai",  
7     "toPlace": "Bangalore",  
8     "journeyDate": "2025-12-10",  
9     "departureTime": "2025-12-10T10:00:00",  
10    "arrivalTime": "2025-12-10T12:00:00",  
11    "airlineName": "Indigo",  
12    "numberOfSeats": 2,  
13    "seatNumbers": "12A,12B",  
14    "mealPreference": "VEG",  
15    "cancelled": false,  
16  }  
17 ]
```

Cancle by pnr:

The screenshot shows a Postman request for the URL `http://localhost:8084/api/v1.0/flight/6930db8025b45b0fe4b8015d`. The method is GET. The response status is 200 OK, with a response time of 14 ms and a body size of 380 B. The response body is a JSON object:

```
1 {
2   "flightId": "6930db8025b45b0fe4b8015d",
3   "departureTime": "2025-12-10T10:00:00",
4   "arrivalTime": "2025-12-10T12:00:00",
5   "airlineName": "Indigo",
6   "priceOneWay": 3500.0,
7   "priceRoundTrip": 0.0,
8   "tripType": "ONE-WAY",
9   "availableSeats": 120,
10  "fromPlace": "Chennai",
11  "toPlace": "Bangalore"
12 }
```

Get flights by id :

The screenshot shows a Postman request for the URL `http://localhost:8084/api/v1.0/flight/6930db8025b45b0fe4b8015d`. The method is GET. The response status is 200 OK, with a response time of 14 ms and a body size of 380 B. The response body is a JSON object:

```
1 {
2   "flightId": "6930db8025b45b0fe4b8015d",
3   "departureTime": "2025-12-10T10:00:00",
4   "arrivalTime": "2025-12-10T12:00:00",
5   "airlineName": "Indigo",
6   "priceOneWay": 3500.0,
7   "priceRoundTrip": 0.0,
8   "tripType": "ONE-WAY",
9   "availableSeats": 120,
10  "fromPlace": "Chennai",
11  "toPlace": "Bangalore"
12 }
```

Jmeter:

Search flights:

The screenshot shows the Apache JMeter interface with a single thread group named "search". The "HTTP Request" sampler is configured with the following parameters:

- Protocol: http
- Server Name or IP: localhost
- Port Number: 8084
- Path: /api/v1/flight/search
- Method: POST
- Body Data:

```
1 "fromPlace": "Hyderabad",
2 "toPlace": "Bangalore",
3 "journeyDate": "2025-12-10",
4 "tripType": "ONE WAY"
5
```

20 samples:

The screenshot shows the Apache JMeter interface with a single thread group named "search". The "Summary Report" listener displays the following results for 20 samples:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
search	20	5	4	9	1.37	0.00%	20.7/sec	8.03	6.78	396.5
TOTAL	20	5	4	9	1.37	0.00%	20.7/sec	8.03	6.78	396.5

5 samples:

The screenshot shows the Apache JMeter interface with a single thread group named "search". The "Summary Report" listener displays the following results for 5 samples:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
search	50	5	4	8	0.87	0.00%	50.7/sec	19.66	16.57	397.4
TOTAL	50	5	4	8	0.87	0.00%	50.7/sec	19.66	16.57	397.4

100 samples:

The screenshot shows the Apache JMeter interface with a single thread group named "search". The "Summary Report" listener displays the following results for 100 samples:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
search	100	5	3	9	0.99	0.00%	100.0/sec	38.75	32.71	396.9
TOTAL	100	5	3	9	0.99	0.00%	100.0/sec	38.75	32.71	396.9

Get flight details by id:

HTTP Request.jmx (C:\Users\SRAVYA\OneDrive\Desktop\Meter scripts\flight-microservices\HTTP Request.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

Test Plan

- Thread Group
 - search
 - View Results Tree
 - Summary Report
 - HTTP Header Manager
- Thread Group
 - get details by id
 - HTTP Header Manager
 - Summary Report
 - View Results Tree

00:00:01 0/120

HTTP Request

Name: get details by id

Comments:

Basic | Advanced

Web Server
Protocol [http]: http
Server Name or IP: localhost
Port Number: 8084

HTTP Request
GET Path: /api/v1.0/flight/6930db8025b45b0fe4b8015d
Content encoding:

Redirect Automatically Follow Redirects Use KeepAlive Use multipart/form-data Browser-compatible headers

Parameters | Body Data | Files Upload |

Sent Parameters With the Request:
Name: Value: URL Encode? Content-Type: Include Equals?

20 samples:

HTTP Request.jmx (C:\Users\SRAVYA\OneDrive\Desktop\Meter scripts\flight-microservices\HTTP Request.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

Test Plan

- Thread Group
 - search
 - View Results Tree
 - Summary Report
 - HTTP Header Manager
- Thread Group
 - get details by id
 - HTTP Header Manager
 - Summary Report
 - View Results Tree

00:00:01 0/120

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file
Filename:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	20	6	5	20	3.22	0.00%	20.9/sec	7.08	4.26	392.0
TOTAL	20	6	5	20	3.22	0.00%	20.9/sec	7.08	4.26	392.0

50 samples:

HTTP Request.jmx (C:\Users\SRAVYA\OneDrive\Desktop\Meter scripts\flight-microservices\HTTP Request.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

Test Plan

- Thread Group
 - search
 - View Results Tree
 - Summary Report
 - HTTP Header Manager
- Thread Group
 - get details by id
 - HTTP Header Manager
 - Summary Report
 - View Results Tree

00:00:01 0/150

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file
Filename:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
get details by id	50	6	5	13	1.48	0.00%	50.8/sec	19.43	10.61	392.0
TOTAL	50	6	5	13	1.48	0.00%	50.8/sec	19.43	10.61	392.0

100 samples:

HTTP Request.jmx (C:\Users\SRAVYA\OneDrive\Desktop\Meter scripts\flight-microservices\HTTP Request.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

Test Plan

- Thread Group
 - search
 - View Results Tree
 - Summary Report
 - HTTP Header Manager
- Thread Group
 - get details by id
 - HTTP Header Manager
 - Summary Report
 - View Results Tree

00:00:01 0/200

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file
Filename:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
get details by id	100	5	4	15	1.30	0.00%	100.2/sec	38.36	20.94	392.1
TOTAL	100	5	4	15	1.30	0.00%	100.2/sec	38.36	20.94	392.1

